Journal of Computer and System Sciences **63**, 531–541 (2001)
doi:10.1006/jcss.2001.1772, available online at http://www.idealibrary.com on IDEAL®

A Randomized Approximation Scheme for Metric MAX-CUT

CNRS, LRI, bat. 490, Université Paris Sud, 91405 Orsay Cedex, France
E-mail: lalo@lri.fr, kenyon@lri.lri.fr

Received March 3, 1999; revised March 31, 2000

Metric MAX-CUT is the problem of dividing a set of points in metric space into two parts so as to maximize the sum of the distances between points belonging to distinct parts. We show that metric MAX-CUT is NP-complete but has a polynomial time randomized approximation scheme.

1. INTRODUCTION

1.1. Background

MAX-CUT, the problem of finding a 2-partition of the vertices of a (possibly weighted) graph which maximizes the number of edges (or sum of edge weights) across the partition, has recently attracted a lot of attention. It has been known for a long time that this basic optimization problem is $\mathcal{NP}$-hard [13] but has a (straightforward) .5-approximation algorithm [23]. The best approximation in the general case is a recent exciting .87856-approximation algorithm due to Goemans and Williamson [14, 15], building upon previous work [7, 8, 22]. Unfortunately, there is not much room for improvement since the problem is Max-$\mathcal{SNP}$-hard [21], and hence [4] has no ε-approximation scheme if $\mathcal{P} \neq \mathcal{NP}$. Thus one is led to consider restricted versions of MAX-CUT. In [3, 9], polynomial time approximation schemes were presented for dense unweighted graphs, *i.e.* graphs with $\Omega(n^2)$ edges, and in [10], dense weighted instances are dealt with. Related results also appear in [16, 12].

In this paper, which is a development of [11], we focus on metric instances of MAX-CUT, when the vertices correspond to points in metric space, the graph is the complete graph, and edge $\{x, y\}$ has a weight equal to the distance between x and y. Our main result is that metric MAX-CUT has a Polynomial Time Approximation Scheme.

0022-0000/01 $35.00

Metric MAX-CUT is mentioned by Bern and Eppstein in [18, chapter 8], in relation to clustering problems. Following his presentation, we loosely define a 2-clustering problem as given by a set X of points in some metric space and seeking the "best" partition of X into two clusters. To measure the quality of the partition, some criterion is applied to each cluster individually. Some criterions for which polynomial-time algorithms exist are the diameter [6, 17] and the variance [5, 19], but there is no polynomial-time algorithm known for minimizing the sum of pairwise distances, which is equivalent to maximizing the sum of distances between points in different clusters, i.e. to metric MAX-CUT. Thus Bern raises the question of designing an efficient approximation algorithm for metric MAX-CUT, which is answered in this paper. Of course, this does not imply a PTAS for minimizing the sum of pairwise distances within each side of the cut. A PTAS for this problem was recently obtained by Indyk [20].

Our proof uses a reduction of Metric MAX-CUT to dense weighted MAX-CUT.D Dense refers usually to the 0, 1 case. Recall that a family of instances of a problem with $o, 1$ weights is said to be dense if the corresponding graphs have average degree at least cn where c is a constant and n denotes the number of vertices of the instance. A PTAS for dense instances of MAX-CUT where found independently by Arora, Karger and Karpinski [3] and Fernandez de la Vega [9]. Actually, we will reduce metric MAX-CUT to an instance of ordinary MAX-CUT in which the maximum weight exceeds the average weight by at most a constant factor. It is almost immediate to check that the algorithms for dense 0, 1 MAX-CUT work for this case with trivial modifications. We refer to [10] for a more general definition of dense weighted instances.

1.2. The Results

The following theorem is due to Luca Trevisan.

THEOREM 1. *Metric MAX-CUT is $\mathcal{NP}$-complete.*

Proof. The proof is a reduction from MAX-CUT. Consider an instance G of MAX-CUT with n vertices. Create a new graph G' with $2n$ vertices by taking two independent copies G_1 and G_2 of G. Create a new weighted complete graph H with $2n$ vertices by giving weight 2 to every edge which was present in G' and weight 1 to all other edges. H is now a metric graph, and it is easy to see that maximum cuts of H correspond to taking a maximum cut (A, B) of G_1 and the complementary maximum cut (B, A) of G_2. If the maximum cut of G has value v, then the maximum cut of G' has value $2v$ and the maximum cut of H has value $2v+n^2$. This completes the reduction. ▮

We now state our approximation theorem.

THEOREM 2. *Metric MAX-CUT has a Polynomial Time Approximation Scheme (PTAS), i.e. for any given $\varepsilon > 0$, there is a randomized algorithm which takes as input a discrete metric space given by its distance matrix, runs in time polynomial in the size*

of the space, and outputs a bipartition whose value is at least $(1-\varepsilon)$ *times the value of the maximum cut.*

One may contrast this result with the situation for the Traveling Salesman Problem (TSP): as is the case for MAX-CUT, the TSP problem is Max-$\mathcal{SNP}$-hard [21], but even the metric version of TSP is also Max-$\mathcal{SNP}$-hard. On the other hand, the Euclidian version of TSP has a PTAS in small dimension [1, 2] while remaining Max-$\mathcal{SNP}$-hard in high dimension [24].

In the Euclidian case and in small dimension, we have a different algorithm which is also a PTAS. Since it is direct, deterministic, and quite natural, we consider it of independent interest, and present it in Section 2.

1.3. Proof Techniques

In the Euclidian case, the algorithm is based on changing coordinates by moving the origin to the center of gravity of the point set, using polar coordinates, suitable rounding to simplify the point set, and using brute force to solve the simplified instance.

In the general metric case, we will obtain our approximation theorem as a consequence of the following reduction.

THEOREM 3. *Approximating Metric MAX-CUT reduces to approximating Dense MAX-CUT.*

Actually, as mentioned before, we will reduce metric MAX-CUT to an instance of ordinary MAX-CUT in which the maximum weight exceeds the average weight by at most a constant factor.

In fact the main idea of the reduction is the following. Let us first see what problem is raised by a naive adaptation of dense graphs algorithms to metric MAX-CUT. The first step usually consists in taking a constant size sample of the vertices. In the dense graphs setting, all significant vertices have the same number of edges (up to a constant factor), hence contribute the same number of edges to MAX-CUT (up to a constant factor), hence a sample of constant size is sufficient to get a fairly good picture of the whole graph. In the metric setting, the situation is completely different. Outliers (points really far from the rest of the set) may contribute much more to MAX-CUT than other points. A constant size sample is bound to miss the few outliers, and examining the sample will not give good information about MAX-CUT. Thus a naive adaptation of the dense graph algorithm to metric MAX- CUT is doomed. The solution to this problem is simple: the critical observation is that the contribution of a point x to MAX-CUT is roughly proportional to the average distance from x to the rest of the set. Thus in the metric setting one should not use a uniform sample of the set of points, but a biased sample, where the probability of taking x in the sample is proportional to the average distance from x to the rest of the set. This is the key to our algorithm. In practice, we create a "graph of clones" obtained by duplicating each vertex a

number of times proportional to its distance to the rest of the set, and perform a uniform sample on the graph of clones.

The analysis uses two main observations: one, that up to a small constant factor, solving metric MAX-CUT reduces to solving MAX-CUT on the (suitably weighted) graph of clones; two, that this graph is a dense instance of MAX-CUT. Both proofs rely on the triangular inequality.

1.4. Open Problems

Reasonable extensions would include trying the same approach to solve other optimization problems of a similar flavour: bisection and cutting into k parts for example.

1.5. A Few Notations

Throughout the paper, we denote by $d(x, y)$ the distance between two points x and y. X is our set of n points. MAXCUT(X) denotes the value of an optimum cut of X.

2. THE EUCLIDIAN CASE

In the Euclidian case, when the dimension of the underlying space is fixed, a PTAS for MAXCUT can easily be obtained. Here, we describe the PTAS for MAX-CUT in the plane. The cases of higher dimension are completely similar (replacing polar coordinates by spherical coordinates).

2.1. The Algorithm

The algorithm is the following.

Input: A set X of n points in the Euclidian plane.

1. Scale the problem so that the average interpoint distance is equal to 1.
2. Compute $g = \sum_{x \in X} x/n$, the center of gravity of X.
3. If $(d(x, g), \theta(x))$ denote the polar coordinates of x w.r. to g, define the domains

$$D_{r,k} = \left\{ x \in R^2 : \begin{array}{l} \varepsilon(1+\varepsilon)^{r-1} \leqslant d(x, g) < \varepsilon(1+\varepsilon)^r \text{ and} \\ k\pi\varepsilon \leqslant \theta(x) < (k+1)\,\pi\varepsilon \end{array} \right\},$$

where $r \geqslant 1$ and $0 \leqslant k < 2\pi/\varepsilon$. Let

$$D_0 = \{x \in R^2 : d(x, g) < \varepsilon\}.$$

4. Construct a point (multi)set X' obtained by replacing each element of $X \cap D_{r,k}$ by $y_{r,k}$, the point with polar coordinates $d(y_{r,k}, g) = \varepsilon(1+\varepsilon)^{r-1}$ and $\theta(y_{r,k} = k\pi\varepsilon$. Hence $y_{r,k}$ has multiplicity equal to the number of points of $X \cap D_{r,k}$. Moreover, each element of $X \cap D_0$ is replaced by g.
5. Solve MAXCUT on X' by doing exhaustive search on the family of all cuts such that points which have the same coordinates are placed on the same side of the cut.

Output: the corresponding cut of X.

2.2. Analysis of the Running Time

The running time of the algorithm is clearly polynomial, with the possible exception of the exhaustive search. The running time of the exhaustive search is exponential in the number of non-empty domains $D_{r,k}$. The following lemma will help up analyze this quantity.

LEMMA 1. *Let $d_{\max} = \max_{x,y \in X} d(x, y)$ denote the diameter of the point set. Then the sum of all interpoint distances is at least*

$$\sum_{\{x,y\} \subset X} d(x, y) \geqslant (n-1)\, d_{\max}.$$

Proof. Let x_0, y_0 be such that $d(x_0, y_0) = d_{\max}$ is maximum. Let X' be obtained from X by orthogonal projection onto line $(x_0\, y_0)$: this can only decrease distances while keeping $d_{\max}$ unchanged. By definition of $d_{\max}$, all points of X' other than x_0 and y_0 must lie between x_0 and y_0, and it is easy to see that the sum of all distances is minimized when all the points of $X' \setminus \{x_0, y_0\}$ are equal. Then the sum of all interpoint distances is exactly $(n-1)\, d_{\max}$, hence the lemma. ∎

COROLLARY 1. *If the average interpoint distance of X is 1, then the diameter of X is at most $n/2$.*

Thus every point is at distance at most $n/2$ from g. Now, if a domain $D_{r,k}$ contains points of X, it must be the case that $\varepsilon(1+\varepsilon)^{r-1} \leqslant n/2$, or in other words, that $r \leqslant 1 + \log_{1+\varepsilon}(n/2\varepsilon)$. The total number of non-empty domains, including D_0, is than at most $1 + (1 + \log_{1+\varepsilon}(n/2\varepsilon))\, 2\pi/\varepsilon$, and the number of cuts that needs to be examined is thus at most $n^{O(1/\varepsilon^2)}$. This dominates the total running time of the algorithm.

2.3. Analysis of Correctness

In this section, we will show that the cut output by the algorithm is close to optimal.

First, it is easy to see that if x and y are two points of X' which have the same coordinates, then there is a maximum cut of X' which places them on the same side of the cut (otherwise, moving either x or y to the other side would improve the cut). Thus the algorithm does indeed compute MAXCUT(X').

The main question is thus comparing MAXCUT(X') to MAXCUT(X). The idea is that points do not move very far when going from X to X'. In fact, if $x \in X \cap D_{r,k}$, then x is moved by at most the diameter d_r of $D_{r,k}$, which satisfies

$$\begin{aligned} d_r &\leqslant \varepsilon(1+\varepsilon)^r - \varepsilon(1+\varepsilon)^{r-1} + \varepsilon^2(1+\varepsilon)^r \pi \\ &\leqslant 5\varepsilon^2(1+\varepsilon)^{r-1} \\ &\leqslant 5\varepsilon\varepsilon(1+\varepsilon)^{r-1} \\ &\leqslant 5\varepsilon d(x, g). \end{aligned}$$

On the other hand, if $x \in D_0$, then x is moved by at most ε. Clearly, moving one point x at distance δ from its original position does not change the value of the optimum cut by more than $\delta(n-1)$. Thus

$$|\text{MAXCUT}(X') - \text{MAXCUT}(X)| \leqslant \varepsilon n(n-1) + 5\varepsilon(n-1)\sum_x d(x, g).$$

LEMMA 2.

$$\sum_{x \in X} d(x, g) \leqslant n/2.$$

Proof. It is easy to see that

$$d(x, g) \leqslant \frac{1}{n}\sum_{y \in X} d(x, y).$$

In one dimension this is clear; in higher dimension it suffices to perform a orthogonal projection of X onto line (xg), which does not affect the left-hand side and can only decrease the right-hand side. Summing over all x yields the lemma. ∎

Using the lemma, we get

$$|\text{MAXCUT}(X') - \text{MAXCUT}(X)| \leqslant 4\varepsilon n^2.$$

The expected value of a random cut of X is $n(n-1)/4$, and so

$$|\text{MAXCUT}(X') - \text{MAXCUT}(X)| \leqslant 17\varepsilon\text{MAXCUT}(X);$$

hence the algorithm is a PTAS.

3. THE GENERAL METRIC CASE

3.1. Notations and Definitions

We need a few important definitions.

Let X be a set of n points in a metric space. The value of a partition (or cut) (A, B) of X is the sum of the distances between points of A and points of B.

DEFINITION 1. The weight w_x, of $x \in X$ is defined as: $w_x = \sum_{y \in X} d(x, y)$. The total weight W of X is defined as: $W = \sum_{x \in X} w_x$.

Again we assume that the average inter-point distance is equal to 1, i.e., that $W = n(n-1)$.

DEFINITION 2. We define a weighted graph $G' = (X', E')$, where:

— The vertex set X' is the *set of clones*; each point $x \in X$ is cloned to create $\lfloor w_x \rfloor$ identical points of X',

— The edge set E' is defined as follows. For each pair x', y' where x' is a clone of x, and y', a clone of y, there is the edge between x' and y' with weight $e_{x'y'} = d(x, y)/(w_x w_y)$.

3.2. The Algorithm

The algorithm is very simple.

Input: A discrete metric space X of size n with distance function d.

1. Scale d so that $\sum_{x \in X} \sum_{x \in X} d(x, y) = n(n-1)$.
2. Construct the graph of clones $G' = (X', E')$.
3. Apply the dense weighted MAX-CUT approximation scheme from [10] to G', so as to get a cut (A', B') of X'.
4. Construct a cut (A, B) of X by doing the following for each $x \in X$ in an incremental manner:
   - Let $x' \in X'$ be any clone of x, let $a_x = \sum_{z' \in A'} e_{x'z'}$ and $b_x = \sum_{z' \in B'} e_{x'z'}$.
   - If $a_x < b_x$ then put x in A and let move all the clones of x to A', otherwise put x in B and move all the clones of x to B'.

Output: Cut (A, B) of X.

3.3. The Analysis

We will need the following lemma.

LEMMA 3.

$$\forall x \in X, W \leqslant 2nw_x.$$

Proof. By the triangular inequality, $d(y, z) \leqslant d(y, x) + d(x, z)$. Summing over all $y, z \in X$ gives the statement of the lemma. ▮

Since we have scaled the problem to instances where $W = n(n-1)$, the weight of any x is at least $(n-1)/2$ and the lemma below follows immediately.

LEMMA 4.

$$\forall x \in X, \lfloor w_x \rfloor \geqslant w_x(1 - 1/n).$$

The theorem below reduces MAX-CUT on X to MAX-CUT on its graph of clones.

LEMMA 5.

$$MAXCUT(X)\left(1 - \frac{2}{n-1}\right)^2 \leqslant MAXCUT(X').$$

Proof. Consider an optimal cut (L, R) of X, and let (L', R') be the induced cut of X' (where L' contains all the clones of elements of L and R' contains all the clones of elements of R). We have

$$\begin{aligned}
\text{Value}(L', R') &= \sum_{x' \in L',\, y' \in R'} e_{x'y'} \\
&= \sum_{x \in L,\, y \in R} \lfloor w_x \rfloor \lfloor w_y \rfloor \frac{d(x, y)}{w_x w_y} \\
&\leqslant \sum_{x \in L,\, y \in R} d(x, y)\left(1 - \frac{2}{n-1}\right)^2 \\
&= \text{MAXCUT}(X)\left(1 - \frac{2}{n-1}\right)^2,
\end{aligned}$$

where the inequality follows from Lemma 4. Since the optimum cut of X' is at least as good as (L', R'), this proves the theorem. ▮

LEMMA 6. *If (A', B') is any cut of X', and (A, B) is the cut of X obtained from (A', B') by proceeding as in step* 4 *of the algorithm, then the value of (A, B) is greater than or equal to the value of (A', B').*

Proof. Let A'_x denote the set of clones of x which belong to A' and B'_x denote the set of clones of x which belong to B'. Using the notations of the algorithm and assuming that we are in the case $a_x < b_x$ (that is, when the algorithm places x in A), we have

$$\text{Value}(A' \cup B'_x, B' \setminus B'_x) = |B'_x|\,(b_x - a_x) \geqslant 0.$$

Thus the cut $(A' \cup B'_x, B' \setminus B'_x)$ of X' is at least as good as (A', B') and has all clones of x on the same side. Repeating this for all $x \in X$ in an incremental fashion, we obtain a cut (A'', B'') of X' which is at least as good as (A', B') and which has all clones of elements of A in A'' and all clones of elements of B in B''. Them

$$\begin{aligned}\text{Value}(A', B') &\leqslant \text{Value}(A'', B'') \\ &= \sum_{x' \in A'',\, y' \in B''} e_{x'y'} \\ &= \sum_{x \in A,\, y \in B} \lfloor w_x \rfloor \lfloor w_y \rfloor \frac{d(x, y)}{w_x w_y} \\ &\leqslant \text{Value}(A, B),\end{aligned}$$

which concludes the proof of the theorem. ∎

All that remains now is to show that (X', E') is a dense weighted graph.

LEMMA 7.

$$\max_{x', y' \in X'} (e_{x'y'}) \leqslant \frac{4}{(1-2/(n-1))^2} \frac{\sum_{x', y' \in X'} e_{x', y'}}{|X'|^2}.$$

Proof. First note that

$$|X'| = \sum_{x \in X} \lfloor w_x \rfloor \leqslant W.$$

Thus the average value of a random entry of the adjacency matrix E' is

$$\begin{aligned}&\frac{\sum_{x, y \in X} \lfloor w_x \rfloor \lfloor w_y \rfloor d(x, y)/(w_x w_y)}{|X'|^2} \\ \leqslant &\frac{(1-2/(n-1))^2 W}{W^2} \\ = &\frac{(1-2/(n-1))^2}{W},\end{aligned}$$

where we have used Lemma 4.

Now, take an arbitrary entry of E'. Using the triangular inequality, we can bound it as follows:

$$\begin{aligned}\frac{d(x, y)}{w_x w_y} &\leqslant \frac{\frac{1}{n}\sum_z (d(x, z)+d(z, y))}{w_x w_y} \\ &= \frac{1}{n}\,\frac{w_x+w_y}{w_x w_y} \\ &= \frac{1}{nw_y}+\frac{1}{nw_x} \\ &\leqslant \frac{4}{W},\end{aligned}$$

where the last inequality follows from Lemma 3.

This implies the theorem. ∎

ACKNOWLEDGMENTS

We thank Luca Trevisan for proving Theorem 1 and accepting that its included in this paper. We also thank Marek Karpinski, Frédéric Magniez, and Richard Kenyon for fruitful discussions.

REFERENCES

1. S. Arora, Polynomial time approximation schemes for Euclidean TSP and other geometric problems, *in* "37th Annual Symposium on Foundations of Computer Science," pp. 2–11, Burlington, VT, 14–16 Oct. 1996, IEEE, to appear in *J. Assoc. Comput. Mach.*.
2. S. Arora, Nearly linear time approximation schemes for Euclidean TSP and other geometric problems, *in* "38th Annual Symposium on Foundations of Computer Science," pp. 554–563, Miami Beach, FL, 20–22 Oct. 1997, IEEE.
3. S. Arora, D. Karger, and M. Karpinski, Approximation schemes for dense instances of NP-hard problems, *in* "Proc. 27th ACM Symposium on the Theory of Computing (STOC)," pp. 284–293, May 1995.
4. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verification and the hardness of approximation problems, *J. Assoc. Comput. Mach.* **45** (1998), 501–555.
5. E. Boros and P. Hammer, On clustering problems with connected optima in Euclidean spaces, *Discrete Math.* **75** (1989), 81–88.
6. V. Capoyleas, G. Rote, and G. Woeginger, Geometric clusterings, *J. Algorithms* **12** (1991), 341–356.
7. C. Delorme and S. Poljak, Combinatorial properties and the complexity of a MAX-CUT approximation, *Eur. J. Combin.* **14** (1993), 313–333.
8. C. Delorme and S. Poljak, Laplacian eigenvalues and the maximum cut problem, *Math. Program.* **62** (1993), 557–574.
9. W. F. de la Vega, MAX-CUT has a randomized approximation scheme in dense graphs, *Random Structures Algorithms* **8** (1996), 187–198.

10. W. F. de la Vega and M. Karpinski, Polynomial time approximation of dense weighted instances of MAX-CUT, *Electronic Colloquium on Computational Complexity Report TR98–064*, 1998.
11. W. F. de la Vega and C. Kenyon, A randomized approximation scheme for metric MAX-CUT, *in* "Proc. of the 39th Annual IEEE Symposium on Foundations of Computer Science," pp. 468–471, November 1998.
12. A. Frieze and R. Kannan, Quick approximations to matrices and applications, *Combinatorica* **19** (1999), 175–220.
13. M. Garey and D. Johnson, "Computers and Intractability," Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
14. M. X. Goemans and D. P. Williamson, .878- approximation algorithms for MAX-CUT and MAX-2SAT, *in* "Proc. 26th Symposium on the Theory of Computing (STOC)," pp. 422–431, May 1994.
15. M. X. Goemans and D. P. Williamson, Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming, *J. Assoc. Comput. Mach.* **42** (1995), 1115–1145.
16. O. Goldreich, S. Goldwasser, and D. Ron, Property testing and its connection to learning and approximation, *in* "Proceedings of 37th Foundations of Computer Science (FOCS)," November 1996.
17. H. Hershberger and S. Suri, Finding tailored partitions, *J. Algorithms* **12** (1991), 431–463.
18. D. S. Hochbaum, "Approximation Algorithms for NP-Hard Problems," PWS, Kent/Boston, 1995.
19. M. Inaba, N. Katoh, and H. lmai, Applications of weighted voronoi diagrams and randomization of variances-based *k*-clustering, *in* "Proceedings of the 10th ACM Symposium on Computational Geometry," pp. 332–339, 1994.
20. P. Indyk, A sublinear-time approximation scheme for clustering in metric spaces, *in* "40th Symposium on Foundations of Computer Science," New York, 1999, IEEE.
21. C. Papadimitriou and M. Yannakakis, Optimization, approximation and complexity classes, *J. Comput. System Sci.* **43** (1991), 425–440.
22. S. Poljak and F. Rendl, Solving the MAX-CUT problem using eigenvalues, *Report 91735-OR, Institute fur Diskrete Mathematik, University of Bonn*, 1991.
23. S. Sahni and T. Gonzalez, P-complete approximation problems, *J. Assoc. Comput. Mach.* **23** (1976), 555–565.
24. L. Trevisan, When Hamming meets Euclid: The approximability of geometric TSP and MST (extended abstract), *in* "Proceedings of the Twenty-Ninth Annual Symposium on Theory of Computing," pp. 21–29, El Paso, Texas, 4–6 May 1997.